Paper No. 21

UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

_____

Ex parte ROY M. STEVENS

_____

Appeal No. 2002-0727
Application No. 08/869,878

_____

ON BRIEF

_____

Before JEFFREY T. SMITH, PAWLIKOWSKI, and MOORE, *Administrative Patent Judges*.

MOORE, *Administrative Patent Judge.*

DECISION ON APPEAL

This is an appeal under 35 U.S.C. § 134 from the final rejection of claims 1-10, all of the pending claims of this application.

REPRESENTATIVE CLAIMS

The appellants have indicated (Brief, page 4) that, for the purposes of this appeal, the claims will stand or fall together. Consistent with this indication, Appellant has made no separate arguments with respect to the remaining claims. Accordingly, all the claims will stand or fall together, and we will select claims 1 and 6, the independent claims, as representative of all of the

claims on appeal. Note In re Dance, 160 F.3d 1339, 1340 n.2, 48 USPQ2d 1635, 1636 n.2 (Fed. Cir. 1998); In re King, 801 F.2d 1324, 1325, 231 USPQ 136, 137 (Fed. Cir. 1986); In re Sernaker, 702 F.2d 989, 991, 217 USPQ 1, 3 (Fed. Cir. 1983). They read as reproduced below.

1. In a computer system including a plurality of processors, a main memory and a cache memory, a method for managing the cache memory comprising the steps of:

(a) dividing said cache memory into a plurality of regions;

(b) associating each of said processors with a different one of said regions;

(c) generating an access address to said main memory that contains data desired by one of said processors;

(d) determining if a copy of said data resides in said cache memory;

(e) providing access to said copy of said data residing in said cache memory if said data resides in any region within said cache memory; and

(f) copying said data from said main memory into the region of said cache memory associated with said one of said processors if a copy of said data does not reside in any region within said cache memory.

6. An apparatus for accelerating the access speed of a main memory, comprising:

(a) a cache memory including a plurality of regions, said cache memory is shared by a plurality of processors, each of said processors is associated with one of said regions;

(b) means for generating an address access that contains data desired by one of said processors; and

(c) means for determining if a copy of said data resides in said cache memory;

(e) means for providing access to said copy of said data residing in said cache memory if said copy of said data resides in any region within said cache memory; and

(f) means for copying said data from said main memory into the region of said cache memory associated with said one of said processors if a copy of said data does not reside in any region within said cache memory.

The References

In rejecting the claims under 35 U.S.C. § 103(a), the examiner relies upon the following references:

Brenza	4,905,141	Feb. 27, 1990
Pierce et al. (Pierce)	5,584,017	Dec. 10, 1996

The Rejection

Claims 1-10 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Brenza in view of Pierce.

The Invention

The invention relates to an apparatus and method for speeding up the access to data housed in a computer by avoiding cross-thrashing of data in a cache memory. (Specification, page 2, lines 12-16 and 18-22). The cache region is divided into a plurality of regions which can be accessed by any of a plurality of processors desiring data. However, when data is not found within the cache memory, each processor can only cause allocation

of data to its respectively assigned region. (Appeal Brief, page 3, lines 1-9). Further details of the claimed invention are as found in claims 1 and 6 reproduced above.

The Rejection of Claims 1-10 Under 35 U.S.C. § 103 (a)

The examiner has found that Brenza discloses a computer system having a plurality of processors, a main memory, and a cache memory, wherein the cache memory is managed by partitioning (Examiner's Answer, page 4, lines 16-21). The examiner has further found that Brenza discloses determining if a copy of desired data resides in the cache memory; providing access if it does; copying the data from the main memory to the cache memory if it does not. (Examiner's Answer, page 4, line 21 - page 5, line 7).

The examiner has additionally found that Pierce discloses cache control wherein snoop cycles can be inhibited by controlling processor access to memory locations. (Examiner's Answer, page 5, lines 13-20).

The examiner thus concludes that it would have been obvious to use Pierce's association of each processor with a specific cache memory in order to improve system efficiency (Examiner's Answer, page 6, lines 7-18).

The appellant, on the other hand, urges that Pierce does not teach a partitioned cache memory as stated by the examiner. Rather, it is said to teach a separate local cache memory associated with each processor, and a common memory shared by all processors but having partitions accessible only by certain processors. (Appeal Brief, page 6, lines 8-17). The local cache memories are not partitioned, and each cache memory is accessible only to its respective processor (Id., page 6, lines 18-20).

After a thorough review of Brenza and Pierce, we find ourselves in agreement with the appellant that a prima facie case of obviousness has not been established by the examiner.

While we agree with the examiner that Brenza teaches a partitioned cache memory (figure 2, reference numeral 50), we note that access to that partitioned cache is controlled by address switches 58 and 52. Each partition operates independently and in parallel, and may execute a store or fetch operation on each machine cycle (column 3, lines 51-53). The partitions are designed around the number of set-associative bins in the computer design (column 3, lines 59-61) and ideally the number of partitions will equal the number of data ports (column 8, lines 22-29). A partition look aside table (PLAT) identifies the partition in which data is located and sends data requests to that partition (column 7, lines 13-18). This enables the cache to

execute a combination of stores and fetches independently and in parallel (column 7, lines 46-57) and provides fault tolerance in the event of chip failure (column 17, line 51 – column 18, line 27).

We do not see any disclosure of assigning partitions to unique processors for storage operations, although there is a discussion at column 17 relating to read-only data storage.

Pierce teaches providing each of a plurality of processors with its own local cache (column 5, lines 35-47). The processors share a common memory which has portions assigned to each of the processors (column 5, lines 53-57). This is said to enhance data integrity and reduce the so-called snoop operation which checks to make sure processors do not maintain separate local caches of the same information which may have been separately manipulated. (column 5, lines 60-61).

However, we fail to see (1) why one of ordinary skill in the art would replace the shared cache of Brenza with the separate caches of Pierce, and (2) even were one so motivated, how the combination of Brenza and Pierce would yield the claimed subject matter.

As to the first point, Pierce teaches using completely separate caches for each processor and assigned common memory portions to reduce the snooping operation, which is inconsistent

with the instantly claimed cache regions accessible for the claimed read operation, but inaccessible for a write operation. Furthermore, combining the separate caches of Pierce into Brenza would appear to defeat the stated goal of the partitioned cache of Brenza being able to operate independently and in parallel.[1] (column 7, lines 46-48).

Furthermore, we observe that the shared memory of Pierce is utilized "as a source for processor instructions and a source/destination for processed information" (column 5, lines 42-43). It does not appear to have a local cache function.

Therefore, even were there sufficient motivation to arrive at the combination, we do not see how the claimed invention would result. The separate caches for each processor are not mutually accessible to each processor, even if the shared memory might be. Both claims 1 and 6 require that access to data must be provided if the copy of the desired data resides in any region within the cache memory. In the absence of this element from the combination, we find a prima facie case of obviousness has not been made out.

Consequently, we are constrained to reverse this rejection.

1 Indeed, this would appear to invite the problem of snoop, which Pierce is

Summary of Decision

The rejection of claims 1-10 under 35 U.S.C. § 103(a) as being unpatentable over Brenza in view of Pierce is reversed.

REVERSED

JEFFREY T. SMITH Administrative Patent Judge	)	
BEVERLY A. PAWLIKOWSKI Administrative Patent Judge	)	BOARD OF PATENT APPEALS AND INTERFERENCES
JAMES T. MOORE Administrative Patent Judge	)	

JTM/ki

designed to solve by splitting off the caches to each processor.

Appeal No. 2002-0727
Application No. 08/869,878

JAMES M. STOVER
NCR CORPORATION
1700 SOUTH PATTERSON BLVD, WHQ4
DAYTON OH 45479